

Research Article

Fast Prediction with Sparse Multikernel LS-SVR Using Multiple Relevant Time Series and Its Application in Avionics System

Yang M. Guo,¹ Pei He,¹ Xiang T. Wang,¹ Ya F. Zheng,¹ Chong Liu,² and Xiao B. Cai³

¹School of Computer Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China

²School of Software and Microelectronics, Northwestern Polytechnical University, Xi'an 710072, China

³Science and Technology Commission, Aviation Industry Corporation of China, Beijing 100068, China

Correspondence should be addressed to Yang M. Guo; yangming_g@nwpu.edu.cn

Received 3 July 2014; Revised 8 October 2014; Accepted 24 November 2014

Academic Editor: Enrico Zio

Copyright © 2015 Yang M. Guo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Health trend prediction is critical to ensure the safe operation of highly reliable systems. However, complex systems often present complex dynamic behaviors and uncertainty, which makes it difficult to develop a precise physical prediction model. Therefore, time series is often used for prediction in this case. In this paper, in order to obtain better prediction accuracy in shorter computation time, we propose a new scheme which utilizes multiple relevant time series to enhance the completeness of the information and adopts a prediction model based on least squares support vector regression (LS-SVR) to perform prediction. In the scheme, we apply two innovative ways to overcome the drawbacks of the reported approaches. One is to remove certain support vectors by measuring the linear correlation to increase sparseness of LS-SVR; the other one is to determine the linear combination weights of multiple kernels by calculating the root mean squared error of each basis kernel. The results of prediction experiments indicate preliminarily that the proposed method is an effective approach for its good prediction accuracy and low computation time, and it is a valuable method in applications.

1. Introduction

Over the past decade, many diagnosis methodologies have been applied successfully in the domain of the reliability and safety of systems [1–7]. However, with the development of prognosis and health management (PHM), health condition diagnosis and prognosis technology has become the key technique for enhancement of reliability and safety of complex systems, especially for high reliability systems. However, these complex systems often exhibit multiple dynamic evolution behaviors, multilevel structures, and incomplete or uncertain information. Therefore, it is difficult to establish an accurate physical model. The time series analysis methods are often utilized to analyze and predict their health conditions in practice [7–10].

Numerous methods have been reported to solve these problems, such as artificial neural networks (ANN) and support vector regression (SVR) [5, 6, 8, 10–14]. ANN has been used in several fields due to its universal approximation

property. However, it may fall into local minimum traps and there is difficulty in determining the hidden layer size and the learning rate [15–17]. On the other hand, SVR aims at the global optimum and exhibits better prediction accuracy due to its implementation of the structural risk minimization principle [18, 19].

SVR is an effective methodology for handling function estimation problems [13, 14, 20]. SVR is performed endogenously as a part of the optimization problem which has a unique solution of a (convex) quadratic programming (QP) problem. The solution of the QP problem provides the necessary information for choosing the most important data points, known as support vectors, among all the sample data. Based on the SVR formulation, support vectors uniquely define the estimated regression function.

The application of SVR is initially developed for performing linear regression. The technique called the kernel trick has been extended to handle nonlinear regression applications [18–20]. Firstly, a kernel function $\varphi(x)$ is used to

map the input data x into an arbitrarily high dimensional feature space. Secondly, the linear SVR can be applied to create an approximate linear function in this arbitrarily high dimensional feature space. This way, nonlinear regression in the low dimensional input space corresponds to linear regression in the high dimensional feature space [21].

The complexity of SVR depends not only on the input space dimension, but also on the number of sample data points. Therefore, when the sample data size is large, the quadratic programming (QP) problem becomes more complex, which will cost a lot of computation time. For this reason, least squares support vector regression (LS-SVR) was proposed by Suykens et al. [22]. In LS-SVR, the inequality constrains are replaced by equality constrains. This way, solving a QP is converted into solving linear equations. This will directly reduce training computational complexity. Thus, the training time of LS-SVR is reduced greatly.

LS-SVR has been demonstrated by simulations and applications that it has better prediction results in most cases [23–26]. Moreover, in order to obtain abundant information on the physical system, researchers proposed multiple relevant time series prediction methods which achieve better prediction results than single time series prediction [27–29]. In the practice of health trend fast prediction for high reliability systems via LS-SVR with multiple relevant time series, we also need to overcome some serious drawbacks in reported methods as described below.

- (1) Because all the training sample data are selected as support vectors, this leads to redundant information in the fitting of LS-SVR and lack of sparseness [30]. This drawback may influence the generalization capability of LS-SVR as well as the training complexity, which will increase prediction time, especially for multiple relevant time series.
- (2) The information contained in the sample data is very often incomplete and complex. Thus, LS-SVR with a single kernel may not be sufficient to represent the information contained in data, and it cannot solve complex problems satisfactorily, especially for prediction with multiple relevant time series sample data. Moreover, how to select appropriate kernel functions remains unsolved in theory.

Aiming at multiple relevant time series fast prediction with LS-SVR, we develop a new scheme to overcome the drawbacks and obtain better prediction performance, that is, accurate prediction results and fast prediction simultaneously. In this paper, combining the merits of LS-SVR with a single kernel for multiple relevant time series [27] and LS-SVR with multikernels for single time series [31], we propose a new fast prediction method to improve the prediction performance with multiple relevant time series and multiple basis kernels. This method provides a regression scheme which integrates multiple kernel learning and enforces sparseness from the training sample data. It includes two aspects.

- (1) We apply a simple computational method to determine the combination weights of multiple basis

kernels by calculating the root mean squared error (RMSE) of prediction with each basis kernel.

- (2) We propose to prune the support vectors by judging the linear correlation among the samples in the high dimensional feature space. This scheme makes the information included in the sample data to be amply represented, reduces the number of support vectors, enhances the generalization performance, and shortens the prediction time with better or acceptable prediction accuracy.

The remainder of the paper is organized as follows. Section 2 gives a brief introduction of LS-SVR; Section 3 proposes the scheme in detail which includes pruning support vectors of LS-SVR and computing combination coefficients of multikernel, and then the fast prediction steps are presented; Section 4 shows three experiments and results analysis; and conclusions are given in Section 5.

2. A Brief Introduction of LS-SVR

Given the labeled data set $S = \{(x_i, y_i), i = 1, 2, \dots, n\}$, where $x_i \in R^d$ is the input sample and $y_i \in R$ is the corresponding output label, the main idea of regression based on support vector machines theory is to map the input data to a higher dimensional feature space by means of a nonlinear mapping $\varphi : R^d \rightarrow F$ and solve the linear regression model shown as follows:

$$\hat{y}_i = w^T \varphi(x_i) + b, \quad (1)$$

where \hat{y}_i is the estimation of y_i , f is the regression function, w is the coefficient value, and b is the bias term. The rationale behind the support vector regression (SVR) is to find w and b that optimize the generalization ability of the regressor by minimizing the regularized loss function. The case of SVR is also a convex quadratic programming (QP) problem, and it has a unique optimal solution. But the computation process is complex. So, many researchers proposed modifications of SVR.

The LS-SVR method is a variant of the SVR [11]. Compared with SVR, LS-SVR preserves the following characteristics: the error variable e_i is used to control deviations from the regression function and a squared loss function is used instead of the insensitive loss function. According to the approach of LS-SVR, the model estimate is given by the following optimization problem:

$$\begin{aligned} \min \quad & J(w, b) = \frac{1}{2} \|w\|^2 + \frac{c}{2} \sum_{i=1}^n e_i^2 \\ \text{s.t.} \quad & y_i = w^T \varphi(x_i) + b + e_i \quad i = 1, 2, \dots, n. \end{aligned} \quad (2)$$

The above formulation is nothing but a regression cost function formulated in the feature space defined by the mapping function $\varphi(x)$. Parameter c determines the trade-off between the model complexity and the goodness of fit to

the sample data. In LS-SVR [22], the Lagrangian function is presented as follows:

$$L(w, b, e, \alpha) = \frac{1}{2}w^T w + \frac{1}{2}c \sum_{i=1}^n e_i^2 - \sum_{i=1}^n \alpha_i (w^T \varphi(x_i) + b + e_i - y_i), \quad (3)$$

where α_i is the i th Lagrange multiplier. Because (2) is also a convex function, it is obvious that the Slater constraint qualification holds [32]. Therefore, the optimal solution of (2) satisfies the KKT conditions. The optimality conditions are shown as follows:

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^n \alpha_i \varphi(x_i) = 0 \implies w = \sum_{i=1}^n \alpha_i \varphi(x_i),$$

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^n \alpha_i = 0 \implies \sum_{i=1}^n \alpha_i = 0,$$

$$\frac{\partial L}{\partial \alpha_i} = w^T \varphi(x_i) + b + e_i - y_i = 0 \implies y_i = w^T \varphi(x_i) + b + e_i,$$

$$\frac{\partial L}{\partial e_i} = ce_i - \alpha_i = 0 \implies e_i = \frac{1}{c} \alpha_i. \quad (4)$$

After eliminating e_i and w from (4), the KKT conditions can be expressed as

$$\begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{1}_n^T \\ \mathbf{1}_n & \mathbf{K} + \frac{\mathbf{I}}{c} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \mathbf{y} \end{bmatrix}, \quad (5)$$

where $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_n]^T$, $\mathbf{K}_{i,j} = k(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$, $\mathbf{1}_n$ is an n -dimensional vector of all ones, \mathbf{I} is a unity matrix, and $\mathbf{y} = [y_1, y_2, \dots, y_n]^T$. Equation (5) can be factorized into a positive definite system [33].

Let $\mathbf{H} = \mathbf{K} + \mathbf{I}/c$, and we get the following equations from (5):

$$\mathbf{1}_n^T \alpha = 0, \quad (6)$$

$$\mathbf{1}_n b + \mathbf{H} \alpha = \mathbf{y}. \quad (7)$$

Multiplying $\mathbf{1}_n^T \mathbf{H}^{-1}$ to both sides of (7), we get

$$\mathbf{1}_n^T \mathbf{H}^{-1} \mathbf{1}_n b + \mathbf{1}_n^T \mathbf{H}^{-1} \mathbf{H} \alpha = \mathbf{1}_n^T \mathbf{H}^{-1} \mathbf{y}. \quad (8)$$

According to (6), (7), and (8), we have

$$\mathbf{1}_n^T \mathbf{H}^{-1} \mathbf{1}_n b = \mathbf{1}_n^T \mathbf{H}^{-1} \mathbf{y}. \quad (9)$$

The Lagrange dual variables α and the bias term b are then obtained solely by

$$\alpha = \mathbf{H}^{-1} (\mathbf{y} - \mathbf{1}_n b), \quad (10)$$

$$b = \mathbf{1}_n^T \mathbf{H}^{-1} \mathbf{y} (\mathbf{1}_n^T \mathbf{H}^{-1} \mathbf{1}_n)^{-1}.$$

Any unlabeled input x can be subsequently estimated by the following function:

$$\hat{y}(x) = \sum_{i=1}^n \alpha_i \varphi(x_i)^T \varphi(x) + b = \sum_{i=1}^n \alpha_i k(x_i, x) + b. \quad (11)$$

For multiple relevant time series, we can consider all sample data at the same time point as one multidimensional vector, which means the input will become multidimensional vectors.

3. Proposed Fast Prediction Method for Multiple Relevant Time Series

In order to obtain better prediction performance, we use multiple relevant time series and multiple kernel learning (MKL) to fully utilize the information contained in data. In this section, we propose an improved scheme with multikernel LS-SVR to accomplish fast prediction. This scheme utilizes new approaches to compute the combination coefficients of multikernel and to decrease the number of support vectors.

3.1. Combination Coefficients of Multiple Kernels. The kernel function and the corresponding kernel parameters are the key issues affecting the model prediction accuracy. An effective kernel function should represent sample data adaptively. General kernel methods use a single kernel function and choose the same corresponding parameter for the whole sample data set. However, the distribution of the sample data in a different mapping space is different. So, MKL was proposed by Lanckriet et al. [34].

MKL is an active research topic in the field of machine learning. It provides a more flexible framework than a single kernel and mines information in data more adaptively and effectively, especially in improving performance of the regression function.

In MKL framework, a combined kernel function is defined as the weighted sum of the individual basis kernels. MKL aims to optimize kernel weights while training the SVR-based methods, such as LS-SVR with multikernel [28, 31, 35]. Though researchers have proposed a variety of methods of integrating multiple kernels from three aspects, which are the composite kernels, the multiscale kernels, and the infinite kernels [35], linear convex combination of basis kernels is still one of the most frequently used methods. For this method, each basis kernel can exploit the full set of features or just use a subset of features. In this paper, using the equations provided by Sonnenburg et al. [36], we consider the combined kernel expressed as follows:

$$k(x_p, x_q) = \sum_{j=1}^m \mu_j k_j(x_p, x_q) \quad (12)$$

$$\text{s.t.} \begin{cases} \sum_{j=1}^m \mu_j = 1 \\ \mu_j \geq 0 \quad j = 1, 2, \dots, m, \end{cases}$$

where m is the number of basis kernels and μ_j is the combining weight for the j th basis kernel. According to

the properties of kernel functions [37], matrix k is symmetric positive semidefinite; that is, $k \geq 0$. We need to normalize all kernel matrices k_j by replacing $k_j(x_p, x_q)$ with the following equation to get unit diagonal matrices:

$$k_j(x_p, x_q) \rightarrow \frac{k_j(x_p, x_q)}{\sqrt{k_j(x_p, x_p)k_j(x_q, x_q)}}. \quad (13)$$

Based on MKL, the LS-SVR optimization problem with the new k matrix derived from (2)–(4) is represented as follows:

$$\min_{w,b,e} \max_{\alpha} L(w, b, e, \alpha) = \max_{\alpha} \min_{w,b,e} L(w, b, e, \alpha). \quad (14)$$

We then transform the new kernel into the matrix form $\mathbf{K} = \sum_{j=1}^m \mu_j \mathbf{K}_j$. When the weights μ_j are constrained to be nonnegative and \mathbf{K}_j are positive semidefinite, the constraint $\sum_{j=1}^m \mu_j \mathbf{K}_j \geq 0$ is satisfied automatically. In this case, with respect to the corresponding parameters motivated by Lanckriet et al. [34] and Ye et al. [38], the solution of (14) can be presented as follows:

$$\begin{aligned} \min_{\mu, \kappa} \quad & \frac{1}{2} (\mathbf{y} + \kappa \mathbf{1}_n)^T \left(\sum_{j=1}^m \mu_j \mathbf{K}_j + \frac{\mathbf{I}}{c} \right)^{-1} (\mathbf{y} + \kappa \mathbf{1}_n) \\ \text{s.t.} \quad & \sum_{j=1}^m \mu_j \mathbf{K}_j \geq 0, \quad \sum_{j=1}^m \mu_j = 1. \end{aligned} \quad (15)$$

Using the Schur complement lemma [32, 39], (15) can be cast to the form of semidefinite program (SDP), and then we reduce the SDP under strict constraints to a quadratically constrained quadratic program (QCQP). The objective function of QCQP is convex in μ and α . In other words, the minimization problem is strictly feasible in μ , and the maximization problem is feasible in α [34, 40]. Such a QCQP problem can be solved efficiently by the interior point methods. The obtained dual variables can be used to fix the optimal kernel coefficients.

For MKL, with linear convex combination of basis kernels, the key is to obtain the optimal combining weights μ_j . Some researchers proposed to simultaneously optimize both the combining weights μ_j and the parameters of LS-SVR, for example, regularization of the parameter c and the basis kernels parameters and so on. Now, the more common way is to adopt optimization software packages, such as MOSEK [41], which can solve the primal and dual problems simultaneously using the interior point methods. But all these solution methods always require much computation time and are complex for applications in practice.

Because the combination of the multiple kernels is a linear combination in this paper, the realistic application will be taken into full account; here, we propose a simple approach to determine the combination weights of the new kernel by calculating the root mean squared error (RMSE) of prediction using each single basis kernel; that is, a smaller RMSE value will result in a bigger weight value.

That the prediction value has less relative error means that the LS-SVR model with the basis kernel is a better model.

Thus, the RMSE of prediction is used as the evaluation criterion. The RMSE of the multiple variables prediction is defined as follows:

$$\delta_{\text{RMSE}} = \sqrt{\frac{1}{nr} \sum_{i=1}^r \sum_{k=1}^n (y_i(k) - \hat{y}_i(k))^2}, \quad (16)$$

where r is the number of relevant variables, n is the number of the original training sample data points, and $y_i(k)$ and $\hat{y}_i(k)$ are the prediction and the actual values, respectively. The combination weights μ_j of the multiple kernels are computed as follows:

$$\mu_j = \frac{\sum_{i=1}^m \delta_i - \delta_j}{(m-1) \sum_{i=1}^m \delta_i}, \quad (17)$$

where δ_j is the prediction RMSE of the j th kernel, $\sum_{i=1}^m \delta_i$ is the sum RMSE of all basis kernels, and $\sum_{i=1}^m \delta_i - \delta_j$ presents the contribution of the j th kernel.

3.2. Linear Correlation-Based Method of Pruning Support Vectors. LS-SVR has a major drawback that is lack of sparseness [42] because almost each sample data point will be a support vector. In order to get a sparse solution, several methods have been proposed. For example, Suykens et al. [43] proposed a pruning scheme based on support vector spectrum that prunes those support vectors with smaller Lagrange multiplier values. de Kruif and de Vries [42] introduced a procedure of pruning according to the smallest approximation error when the support vector is omitted. Hoegaerts et al. [44] improved the method proposed by Kruif et al. They suggested a variant that improved the performance significantly and assessed its relative performance compared with two other subset selection schemes. Keerthi and Shevade [45] extended the well-known SMO algorithm to LS-SVM to solve problems with large training samples. Based on this study, Zeng and Chen [46] proposed a new pruning algorithm for sparse LS-SVR via SMO algorithm. Jiao et al. [47] presented two fast sparse approximation schemes for LS-SVR.

The basic idea of the many approaches reviewed above is to find the support vector with the smaller indicator value and remove the corresponding training sample. It is obvious that the idea is simple, but all the reported schemes require computation duplication in solving linear equations [48, 49]. For this reason, these schemes are not suitable for fast prediction. Thus, Yaakov et al. [50] and Cawley and Talbot [51] presented a more efficient scheme by evaluating the linear correlation among the samples in the mapping space. The merits of [48–51] are that the linear correlation-based sparseness strategy ensures the information in the data to be maximally retained and the sparseness prediction model improves the generalization capability to improve the model's adaptability to noisy sample data. So, the proposed method bases on [48–51] and uses the following idea.

Suppose that $\varphi(x_k)$ is one of the n support vectors in the high dimensional feature space. If and only if $\lambda_i = 0$ ($i \neq k$), the following equation holds:

$$\varphi(x_k) \neq \sum_{i=1, i \neq k}^n \lambda_i \varphi(x_i). \quad (18)$$

Here, x_k is called base vector (BV) and the set consisting of x_k is called base vector set (BVS). In order to obtain sparseness in LS-SVR, any support vector linearly related to the BV can be deleted. Assume that there are n support vectors in the sample data and the BVS consists of l support vectors ($l < n$). If the high dimensional mapping $\varphi(x_t)$ of the $(l + 1)$ th sample x_t cannot be represented by a linear combination of the l support vectors, it will be added into the BV; otherwise, it does not join the BVS.

In many real cases, the new sample data may not be absolutely linearly dependent on the existing support vectors. However, it is very similar to a linear combination of the existing support vectors; that is, for sample x_t , we will be content with finding coefficients $\lambda_{t,i}$ with at least one nonzero element satisfying the approximate linear dependence condition

$$\delta_t = \left\| \sum_{i=1}^l \lambda_{t,i} \varphi(x_i) - \varphi(x_t) \right\|^2 \leq \nu, \quad (19)$$

where ν is a small positive constant. Then, [48, 50] rewrote (19) into the following optimization problem:

$$\begin{aligned} \min_{\lambda_t} \quad & \delta_t = \sum_{i=1}^l \sum_{j=1}^l \lambda_{t,i} \varphi(x_i) \lambda_{t,j} \varphi(x_j)^T \\ & - 2\varphi(x_t) \sum_{i=1}^l \lambda_{t,i} \varphi(x_i)^T + \varphi(x_t) \varphi(x_t)^T \\ & = \sum_{i=1}^l \sum_{j=1}^l \lambda_{t,i} \lambda_{t,j} k(x_i, x_j) \\ & - 2 \sum_{i=1}^l \lambda_{t,i} k(x_t, x_i) + k(x_t, x_t). \end{aligned} \quad (20)$$

By minimizing the left-hand side of (20), we can obtain the linear correlation coefficients $\lambda_t = [\lambda_{t,1}, \lambda_{t,2}, \dots, \lambda_{t,l}]^T$ simultaneously.

To sacrifice the optimality of λ_t in return for a reduction in the size of the sample data, a l_2 norm regularization term of the form $\gamma \|\lambda_t\|^2$ is added to the minimization problem defined in (20), ending up with the following optimization problem:

$$\begin{aligned} \min_{\lambda_t} \quad & \delta_t = \sum_{i=1}^l \sum_{j=1}^l \lambda_{t,i} \lambda_{t,j} k(x_i, x_j) \\ & - 2 \sum_{i=1}^l \lambda_{t,i} k(x_t, x_i) + k(x_t, x_t) + \gamma \|\lambda_t\|^2, \end{aligned} \quad (21)$$

parameters: l, c , support vectors set $\{x_1, x_2, \dots, x_l, \dots, x_n\}$
Initialize: $l = 2$, c equals to the LS-SVR parameter c
(1) choose the initial BV: x_1 and x_2 ;
(2) establish the initial BVS: $\{x_1, x_2\}$;
(3) get new support vector x_i ;
(4) approximate the linear dependence compute via (23)
 if $\delta'_t > 0$
 $l = l + 1$,
 $x_l = x_i$,
 add x_i into the BVS,
 else
 drop x_i ,
 $l = l$
(5) go back to (3).

PSEUDOCODE 1: Pseudocode of the proposed pruning method.

where γ is a positive constant coefficient. It is obvious that the sparseness level and the model accuracy are controlled by the parameters ν and γ . But no rules have been reported in the literature for the selection of these two parameters.

In this paper, we select $\nu = \|\lambda_t\|^2 / lc^2$, and then we propose a new sparseness method, which is described as follows:

$$\delta_t = \left\| \sum_{i=1}^l \lambda_{t,i} \varphi(x_i) - \varphi(x_t) \right\|^2 \leq \frac{\|\lambda_t\|^2}{lc^2}. \quad (22)$$

Equation (22) can be solved as the following optimization problem:

$$\begin{aligned} \min_{\lambda_t} \quad & \delta'_t = \sum_{i=1}^l \sum_{j=1}^l \lambda_{t,i} \varphi(x_i) \lambda_{t,j} \varphi(x_j)^T \\ & - 2\varphi(x_t) \sum_{i=1}^l \lambda_{t,i} \varphi(x_i)^T + \varphi(x_t) \varphi(x_t)^T - \frac{\|\lambda_t\|^2}{lc^2} \\ & = \sum_{i=1}^l \sum_{j=1}^l \lambda_{t,i} \lambda_{t,j} k(x_i, x_j) \\ & - 2 \sum_{i=1}^l \lambda_{t,i} k(x_t, x_i) + k(x_t, x_t) - \frac{\|\lambda_t\|^2}{lc^2} \\ \text{s.t.} \quad & |\lambda_{t,i}| \leq c, \end{aligned} \quad (23)$$

where c is the same as the LS-SVR parameter c . Solving (23) and if δ'_t is bigger than $\|\lambda_t\|^2 / lc^2$, add x_t into the BVS; otherwise, x_t is dropped.

Following the ideas outlined above, we propose the following approach to prune support vectors. A detailed pseudocode account of the proposed method is given in Pseudocode 1.

Based on the pseudocode, we choose two sample data points corresponding to the two biggest Lagrange multipliers as the initial BV firstly, and then if the new nonlinear mapping $\varphi(x_r)$ depends linearly on the selected support vectors, it can

be dropped; that is, if the new sample data x_r satisfies the relation formula

$$\varphi(x_r) = \sum_{j=1}^m \lambda_j \varphi(x_j), \quad (24)$$

where λ_i is a constant coefficient and m is the number of independent support vectors, then the new data point is dropped. Combining (24) with (11), we have

$$\begin{aligned} \hat{y}(x) &= \sum_{i=1, i \neq r}^n \alpha_i k(x_i, x) + \alpha_r k(x_r, x) + b \\ &= \sum_{i=1, i \neq r}^n \alpha_i \varphi(x_i) \varphi(x) + \alpha_r \varphi(x_r) \varphi(x) + b \\ &= \sum_{i=1, i \neq r}^n \alpha_i \varphi(x_i) \varphi(x) + \alpha_r \sum_{j=1}^m \lambda_j \varphi(x_j) \varphi(x) + b, \end{aligned} \quad (25)$$

where x_j is the sample corresponding to BV. Then, we will get the new regression function with BV, and it can be described by

$$\hat{y}(x) = \sum_{i=1}^m \beta_i k(x_i, x) + b, \quad (26)$$

where β_i is combined with Lagrange dual variable coefficient and linear combination coefficient.

The proposed method combines advantages of removing smaller support vectors and the advantages of [48–51]. It can retain much of the useful information in the sample data. Thus, the prediction accuracy will not be sacrificed too much while the model's generalization ability is improved. Although this processing step would increase the calculation time for the LS-SVR model, the training time will not be increased much because almost each middle parameter's value is already computed and stored in the process of setting up the prediction model, and the prediction time will be reduced because of fewer support vectors. Moreover, the proposed method can avoid a difficult problem in Yaakov's scheme, the parameters selection, because no rules have been reported to fix the problem. Thus, it may be more valuable in practice.

3.3. Prediction Steps of the Proposed Method. By applying the scheme proposed above, we can set up the LS-SVR model with multikernel for multiple relevant time series, called the improved MKLS-SVR (IMKLS-SVR), and we will use this new model to carry out multiple relevant time series prediction. Here, we look at the multiple relevant time series as an integral whole; that is, they are taken as a multidimensional input [52, 53], and the prediction is executed simultaneously.

The prediction steps can be described as follows:

- (1) choose appropriate basis kernels;
- (2) determine μ_j and then obtain the combination kernel;
- (3) compute \mathbf{K} , \mathbf{H} and $\boldsymbol{\alpha}$, b ;

- (4) assess the linear correlation and prune support vectors and compute new BV-based Lagrange multiplier β_i ;
- (5) set up the regression function and perform prediction.

4. Experiments and Result Analysis

In order to examine the prediction efficiency of the proposed method, we provide simulation and application experiments. All the experiments use MatlabR2011b with LS-SVMLab1.8 Toolbox (the software and guide book can be downloaded from <http://www.esat.kuleuven.be/sista/lssvmlab>) under Windows XP operating system.

4.1. Simulation Experiments and Results Analysis. The simulation experiments include two parts. One is to test the proposed method based on linear correlation (Experiment 1) and the other is to test the proposed computing method of multikernel combination coefficients (Experiment 2). All tests are repeated 50 times with the results averaged. The two experiments to test the proposed scheme are presented below in detail.

Experiment 1. We first used LS-SVR to learn the 1-dimensional function $x = \sin(t)/t$ defined in the interval $t \in [0.1, 10]$. We collect 100 values of x as our time series. The data points from 1 to 60 in the time series are taken as the 50 initial training sample data points. The first sample data set consists of points 1 through 11, with the first 10 as the input sample vector and the 11th point as the output. The second sample data set consists of points 2 through 12, with the points 2 through 11 as the input sample vector and the 12th point as the output. This way we have 50 training data points out of the first 60 data points.

In order to test the efficiency of the proposed pruning method, we compare it with the method which Yaakov et al. presented in [50]. Here, the kernel is Gaussian RBF kernel $k(x, y) = \exp(-\|x - y\|^2/2\sigma^2)$ with standard deviation $\sigma = 1$. The other parameters are $\nu = 0.01$ and $\nu = 0.001$. In this experiment, the training time, prediction time, and prediction mean RMSE are compared. The results are shown in Table 1.

Table 1 shows that the proposed method attains good accuracy with less computation time. In particular, it can reduce the prediction time greatly without losing much prediction accuracy. In addition, this method avoids the problem of parameter selection.

Experiment 2. Henon function is a typical time series, and its variables influence each other. Its corresponding differential equations are given as follows:

$$\begin{aligned} x(t+1) &= 1 - ax^2(t) + y(t), \\ y(t+1) &= bx(t). \end{aligned} \quad (27)$$

Let $a = 1.4$ and $b = 0.3$, and points of the initialization of x and y are $[0.1, 0.1]$. 100 data points of variables x and y are

TABLE 1: Prediction results of simulation Experiment 1.

Method	Training time (s)	Prediction time (s)	Mean RMSE
LS-SVR without sparseness	0.001053	0.018954	1.5032
Method of [50]			
$\nu = 0.01$	0.002072	0.009461	1.8673
$\nu = 0.001$	0.003179	0.009241	1.7780
The proposed method	0.002242	0.009407	1.6160

TABLE 2: Prediction results of simulation Experiment 2.

Method	Training time (s)	Prediction time (s)	Mean RMSE	
			x	y
LS-SVR with RBF kernel	1.273911	1.264925	3.1418	4.6412
Method of [31]	2.783566	0.962933	1.4001	0.8322
The Proposed method	1.659715	0.877201	1.5273	1.0125

TABLE 3: Prediction results of application experiment.

Method	Mean RMSE			Mean total prediction time reduced (compare with LS-SVR)
	x	y	z	
LS-SVR of [25]	3.0662	1.5252	2.3811	
MKLS-SVR of [31]	2.4722	0.7416	1.8869	6.18%
The proposed IMKL-SVR	2.5668	0.8651	1.8412	13.83%

selected, respectively, as sample time series. We set the first 80 data points as training samples. We also take any continuous 11 data points of x and y as a sample, where the first 10 data points compose an input sample vector and the last one is the output vector; that is, in this simulation experiment, we have 70 training data points for each variable. And then we predict number 81 to number 100 time series data using the trained model.

In order to test the practicability of the proposed computing method of multikernel combination coefficients, we choose one linear kernel function $k(x, y) = x^T y$ representing global information and two Gaussian RBF functions (σ^2 equal to 5 and 9, resp.) representing local information [54] and establish multikernel LS-SVR, respectively, by two approaches which are the proposed method and the one described in [31]. After determining the new combination weights of the basis kernels, we compare the training time, prediction time, and prediction mean RMSE with LS-SVR (with single Gaussian RBF kernel, $\sigma^2 = 5$) and multikernel LS-SVR. The results are shown in Table 2.

The simulation results in Table 2 show that the proposed method can reduce computation time more than the other two methods, although it does not achieve the best prediction results. The proposed method has good prediction accuracy and requires less prediction time.

4.2. Avionics System Experiment and Results Analysis. In this section, we use a certain circuit of avionics system (shown in Figure 1) to demonstrate the efficiency of the proposed method with multiple relevant time series. The circuit is used

to compare the prediction time and prediction accuracy with the traditional LS-SVR model described in [25] and LS-SVR with multikernel, called MKLS-SVR, shown in [31].

In order to collect the sample data, we measure voltage values at three measuring points named point A, point B, and point C (with pentagram representations, see Figure 1). The acquisition step is 0.1 s with time unit of 4.5 s.

We take the first 30 measured data points group as the training samples. Again each sample set consists of any continuous 11 data points; that is, we have 20 groups of training data. And then we predict number 31 to number 45 time series data. Here, we look at the time series obtained from point A, point B, and point C (see Figure 1) as an integral whole; that is, they are examined as a multidimensional input and do prediction simultaneously. We run 100 times of prediction and take the average of them. Moreover, all the parameters of kernel functions are jointly optimized with the traditional grid search method, where the search range for c and σ^2 is [0.01, 2000], and the prediction performance is measured by RMSE.

Gaussian RBF kernel is adopted as the kernel function of traditional LS-SVR. For MKL-SVR and IMKLS-SVR, we also choose one linear kernel function and two Gaussian RBF functions. The prediction results are shown in Figures 2, 3, and 4 (dimensions omitted) and the mean prediction RMSE for three data sets are reported in Table 3.

Figures 2–4 and Table 3 show that (1) compared with traditional LS-SVR, the proposed method, IMKLS-SVR, has good prediction accuracy. This may be due to the fact that the proposed method uses relevant information fully and maps them in multikernel high dimensional space; (2) in spite

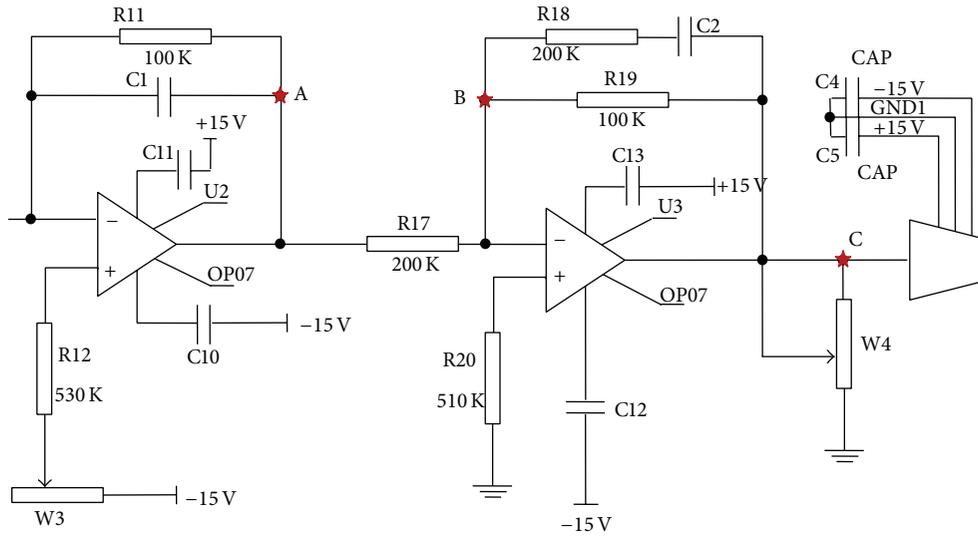


FIGURE 1: Local circuit of avionics system.

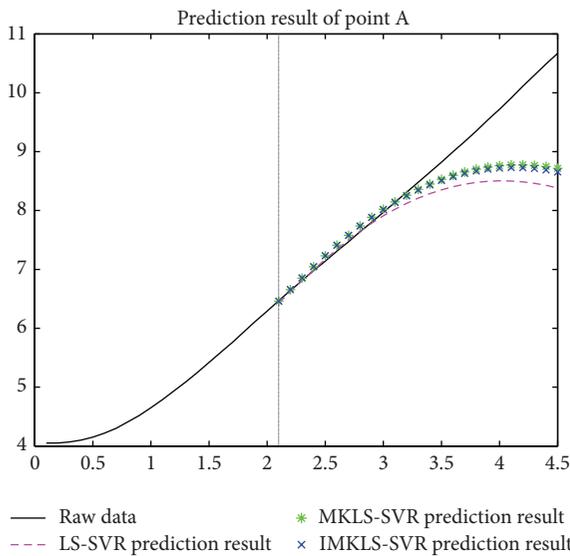


FIGURE 2: Prediction results of data collected from point A (see Figure 1).

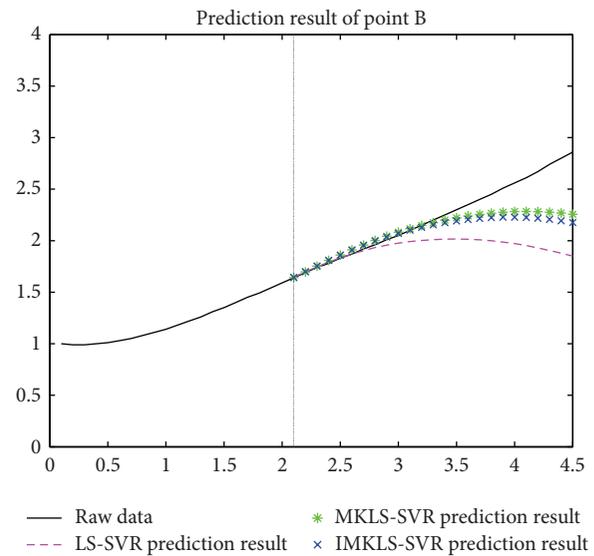


FIGURE 3: Prediction results of data collected from point B (see Figure 1).

of almost the same accuracy with MKLS-SVR, IMKLS-SVR requires less prediction time than MKLS-SVR and LS-SVR, especially than LS-SVR. This may be due to the fact that we have reduced the support vectors of sample data by measuring the linear correlation of the sample data. Thus, IMKLS-SVR is more valuable in application.

5. Conclusions

In this paper, we study the issue of fast prediction with multikernel LS-SVR for multiple relevant time series. We utilize two approaches to overcome the drawbacks of LS-SVR to meet the requirements of good prediction accuracy and less prediction time cost. Firstly, because each time series data

point may have different distribution characteristics, we apply MKL, that is, a linear combination of basis kernels, to enhance the generalization capability of the learning machine and to exploit all discriminative information in the sample data. We determine the linear combination weights by calculating the RMSE of each basis kernel to yield the new kernel. Secondly, we remove the samples by judging the linear correlation to reduce the number of support vectors. This approach can control the loss of useful information of the sample data, reduce computation time, and improve the model's generalization ability simultaneously.

We conducted several experiments to evaluate the proposed methods; we especially utilized a certain prediction application to demonstrate the effectiveness of the proposed

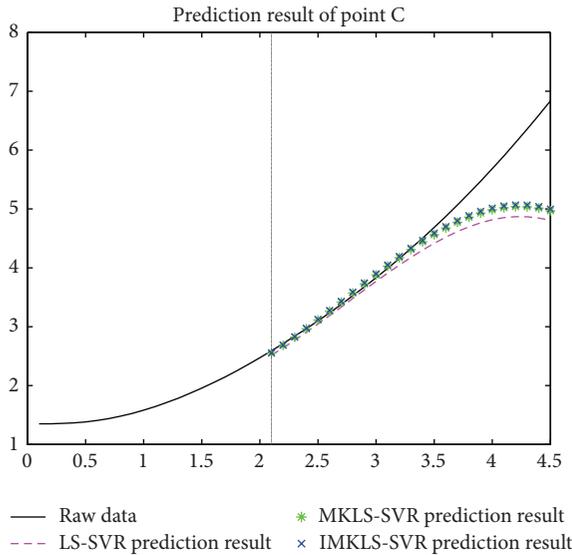


FIGURE 4: Prediction results of data collected from point C (see Figure 1).

method via comparing with the traditional LS-SVR and multikernel LS-SVR. The results show that the proposed prediction scheme has good prediction performance, and it is suitable for multiple relevant time series fast prediction.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by Major Project of Chinese National Programs for Fundamental Research and Development (973 Program), National Natural Science Foundation of China under Grant no. 61371024, Aviation Science Fund of China under Grant no. 2013ZD53051, Aerospace Technology Support Fund of China, the Industry-Academy-Research Project of AVIC (cxy2013XGD14), and the Open Research Project of Guangdong Key Laboratory of Popular High Performance Computers/Shenzhen Key Laboratory of Service Computing and Applications.

References

- [1] Y. Gritli, L. Zarri, C. Rossi, F. Filippetti, G.-A. Capolino, and D. Casadei, "Advanced diagnosis of electrical faults in wound-rotor induction machines," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 9, pp. 4012–4024, 2013.
- [2] X. Jin, M. Zhao, T. W. S. Chow, and M. Pecht, "Motor bearing fault diagnosis using trace ratio linear discriminant analysis," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 5, pp. 2441–2451, 2014.
- [3] T. Boukra, A. Lebaroud, and G. Clerc, "Statistical and neural-network approaches for the classification of induction machine faults using the ambiguity plane representation," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 9, pp. 4034–4042, 2013.
- [4] R. Levy, S. A. Arogeti, and D. Wang, "An integrated approach to mode tracking and diagnosis of hybrid systems," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 4, pp. 2024–2040, 2014.
- [5] Y. Shatnawi and M. Al-Khassaweneh, "Fault diagnosis in internal combustion engines using extension neural network," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 3, pp. 1434–1443, 2014.
- [6] M. D. Prieto, G. Cirrincione, A. G. Espinosa, J. A. Ortega, and H. Henao, "Bearing fault detection by a novel condition-monitoring scheme based on statistical-time features and neural networks," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 8, pp. 3398–3407, 2013.
- [7] E. G. Strangas, S. Aviyente, and S. S. H. Zaidi, "Time-frequency analysis for efficient fault diagnosis and failure prognosis for interior permanent-magnet AC motors," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 12, pp. 4191–4199, 2008.
- [8] A. Soualhi, H. Razik, G. Clerc, and D. D. Doan, "Prognosis of bearing failures using hidden markov models and the adaptive neuro-fuzzy inference system," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 6, pp. 2864–2874, 2014.
- [9] S. S. H. Zaidi, S. Aviyente, M. Salman, K.-K. Shin, and E. G. Strangas, "Prognosis of gear failures in dc starter motors using hidden Markov models," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 5, pp. 1695–1706, 2011.
- [10] M. Qi and G. P. Zhang, "Trend time-series modeling and forecasting with neural networks," *IEEE Transactions on Neural Networks*, vol. 19, no. 5, pp. 808–816, 2008.
- [11] S. Qiu and T. Lane, "A framework for multiple kernel support vector regression and its applications to siRNA efficacy prediction," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 6, no. 2, pp. 190–199, 2009.
- [12] D. Basak, S. Pal, and D. C. Patranabis, "Support vector regression," *Neural Information Processing—Letters and Reviews*, vol. 11, no. 10, pp. 203–224, 2007.
- [13] O. A. Omitaomu, M. K. Jeong, and A. B. Badiru, "Online support vector regression with varying parameters for time-dependent data," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 41, no. 1, pp. 191–197, 2011.
- [14] A. G. Abo-Khalil and D. C. Lee, "MPPT control of wind generation systems based on estimated wind speed using SVR," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 3, pp. 1489–1490, 2008.
- [15] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smol, and V. Vapnik, "Support vector regression machines," *Advances in Neural Information Processing Systems*, vol. 9, pp. 155–161, 1997.
- [16] V. Kecman, *Learning and Soft Computation: Support Vector Machines, Neural Networks, and Fuzzy Logic Models*, The MIT Press, Cambridge, Mass, USA, 2001.
- [17] J. V. Hansen and R. D. Nelson, "Neural networks and traditional time series methods: a synergistic combination in state economic forecasts," *IEEE Transactions on Neural Networks*, vol. 8, no. 4, pp. 863–873, 1997.
- [18] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA, 2nd edition, 1995.
- [19] V. Vapnik, *Statistical Learning Theory*, Wiley-Interscience, New York, NY, USA, 1998.

- [20] A. Smola and B. Schölkopf, "A tutorial on support vector regression," Neuro COLT Technical Report NC-TR-98-030, Royal Holloway College, University of London, 1998.
- [21] K. Müller, A. Smola, G. Ratsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik, *Advance in Kernel Methods: Support Vector Machines*, MIT Press, Cambridge, Mass, USA, 1999.
- [22] J. A. K. Suykens, T. van Gestel, J. de Brabanter, B. de Moor, and J. Vandewalle, *Least Squares Support Vector Machines*, World Scientific Publishing, Singapore, 2002.
- [23] H. B. Guo and X. Q. Guan, "Application of least squares support vector regression in network flow forecasting," in *Proceedings of the 2nd International Conference on Computer Engineering and Technology*, vol. 7, pp. V7-342–V7-345, Chengdu, China, April 2010.
- [24] Y. H. Zhao, P. Zhong, and K. N. Wang, "Application of least squares support vector regression based on time series in prediction of gas," *Journal of Convergence Information Technology*, vol. 6, no. 1, pp. 243–250, 2011.
- [25] J. Qu and M. J. Zuo, "An LSSVR-based algorithm for online system condition prognostics," *Expert Systems with Applications*, vol. 39, no. 5, pp. 6089–6102, 2012.
- [26] Y. M. Guo, C. B. Ran, X. L. Li, and J. Z. Ma, "Adaptive online prediction method based on LS-SVR and its application in an electronic system," *Journal of Zhejiang University SCIENCE C*, vol. 13, no. 12, pp. 881–890, 2012.
- [27] Y. M. Guo, Z. J. Zhai, and H. M. Jiang, "Weighted prediction of multi-parameter chaotic time series using least squares support vector regression (LS-SVR)," *Journal of Northwestern Polytechnical University*, vol. 27, no. 1, pp. 83–86, 2009.
- [28] X. R. Zhang, L. Y. Hu, and Z. S. Wang, "Multiple kernel support vector regression for economic forecasting," in *Proceedings of the 17th International Conference on Management Science and Engineering (ICMSE '10)*, pp. 129–134, Melbourne, Australia, November 2010.
- [29] X. H. Zhao, G. Wang, K. K. Zhao, and D. J. Tan, "On-line least squares support vector machine algorithm in gas prediction," *Mining Science and Technology*, vol. 19, no. 2, pp. 194–198, 2009.
- [30] K. R. Müller, A. J. Smola, G. Ratsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik, "Predicting time series with support vector machines," in *Artificial Neural Networks—ICANN'97*, vol. 1327 of *Lecture Notes in Computer Science*, pp. 999–1004, Springer, Berlin, Germany, 1997.
- [31] G. M. Xu and S. G. Huang, "Runway incursion event forecast model based on LS-SVR with multi-kernel," *Journal of Computers*, vol. 6, no. 7, pp. 1346–1352, 2011.
- [32] L. Jian, Z. Xia, X. Liang, and C. Gao, "Design of a multiple kernel learning algorithm for LS-SVM by convex programming," *Neural Networks*, vol. 24, no. 5, pp. 476–483, 2011.
- [33] F. Ojeda, J. A. K. Suykens, and B. De Moor, "Low rank updated LS-SVM classifiers for fast variable selection," *Neural Networks*, vol. 21, no. 2-3, pp. 437–449, 2008.
- [34] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semi-definite programming," *Journal of Machine Learning Research*, vol. 5, pp. 27–72, 2004.
- [35] H. Q. Wang, F. C. Sun, Y. N. Cai, N. Chen, and L. G. Ding, "Multiple kernel learning methods," *Acta Automatica Sinica*, vol. 36, no. 8, pp. 1037–1050, 2010.
- [36] S. Sonnenburg, G. Räosch, C. Schäfer, and B. Schölkopf, "Large scale multiple kernel learning," *Journal of Machine Learning Research*, vol. 7, pp. 1531–1565, 2006.
- [37] B. Scholkopf and A. Smola, *Learning with Kernels*, MIT Press, Cambridge, Mass, USA, 2001.
- [38] J. P. Ye, S. W. Ji, and J. H. Chen, "Multi-class discriminant kernel learning via convex programming," *Journal of Machine Learning Research*, vol. 9, pp. 719–758, 2008.
- [39] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM Review*, vol. 38, no. 1, pp. 49–95, 1996.
- [40] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004.
- [41] E. D. Andersen and A. D. Andersen, *High Performance Optimization*, vol. 33, Kluwer Academic Publishers, Norewell, Mass, USA, 2000.
- [42] B. J. de Kruif and T. J. A. de Vries, "Pruning error minimization in least squares support vector machines," *IEEE Transactions on Neural Networks*, vol. 14, no. 3, pp. 696–702, 2003.
- [43] J. A. K. Suykens, L. Lukas, and J. Vandewalle, "Sparse approximation using least squares support vector machines," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, vol. 2, pp. 757–760, May 2000.
- [44] L. Hoegaerts, J. A. K. Suykens, J. Vandewalle, and B. de Moor, "A comparison of pruning algorithms for sparse least squares support vector machines," in *Neural Information Processing*, vol. 3316 of *Lecture Notes in Computer Science*, pp. 1247–1253, 2004.
- [45] S. S. Keerthi and S. K. Shevade, "SMO algorithm for least-squares SVM formulations," *Neural Computation*, vol. 15, no. 2, pp. 487–507, 2003.
- [46] X. Y. Zeng and X. W. Chen, "SMO-based pruning methods for sparse least squares support vector machines," *IEEE Transactions on Neural Networks*, vol. 16, no. 6, pp. 1541–1546, 2005.
- [47] L. C. Jiao, L. F. Bo, and L. Wang, "Fast sparse approximation for least squares support vector machine," *IEEE Transactions on Neural Networks*, vol. 18, no. 3, pp. 685–697, 2007.
- [48] S. L. Chen, G. Chen, and H. Xue, "Research on sparsification approach for least squares support vector machine classification," *Computer Engineering*, vol. 37, no. 22, pp. 145–150, 2011.
- [49] A. J. Chen, Z. H. Song, and P. Li, "Modeling method of least squares support vector regression based on vector base learning," *Control Theory and Applications*, vol. 24, no. 1, pp. 1–5, 2007.
- [50] E. Yaakov, M. Shie, and M. Ron, "Sparse online greedy support vector regression," in *Proceedings of the 13th European Conference on Machine Learning*, August 2002.
- [51] G. C. Cawley and N. L. C. Talbot, "Reduced rank kernel ridge regression," *Neural Processing Letters*, vol. 16, no. 3, pp. 293–302, 2002.
- [52] M. Lázaro, I. Santamaría, F. Pérez-Cruz, and A. Artés-Rodríguez, "Support Vector Regression for the simultaneous learning of a multivariate function and its derivatives," *Neurocomputing*, vol. 69, no. 1–3, pp. 42–61, 2005.
- [53] A. Shilton, D. T. H. Lai, and M. Palaniswami, "A division algebraic framework for multidimensional support vector regression," *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, vol. 40, no. 2, pp. 517–528, 2010.
- [54] J. F. Zhang and S. S. Hu, "Chaotic time series prediction based on multi-kernel learning support vector regression," *Acta Physica Sinica*, vol. 57, no. 5, pp. 2708–2713, 2008.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

